Characterizing Hadoop Applications on Microservers for Performance and Energy Efficiency Optimizations

Maria Malik[1], Avesta Sasan[1], Rajiv Joshi[3], Setareh Rafatirah[2], Houman Homayoun[1]
[1]Department of Electrical and Computer Engineering, [2]Department of Information Sciences and Technology
George Mason University, Fairfax, VA, USA, {mmalik9, asasan, srafatir, hhomayou}@gmu.edu
[3]IBM Research, Yorktown Heights, NY, {rvjoshi@us.ibm.com}

Abstract-The traditional low-power embedded processors such as Atom and ARM are entering the high-performance server market. At the same time, as the size of data grows, emerging Big Data applications require more and more server computational power that yields challenges to process data energy-efficiently using current high performance server architectures. Furthermore, physical design constraints, such as power and density have become the dominant limiting factor for scaling out servers. Numerous big data applications rely on using the Hadoop MapReduce framework to perform their analysis on large-scale datasets. Since Hadoop configuration parameters as well as architecture parameters directly affect the MapReduce job performance and energy-efficiency, system and architecture level parameters tuning is vital to maximize the energy efficiency. In this work, through methodical investigation of performance and power measurements, we demonstrate how the interplay among various Hadoop configurations and system and architecture level parameters affect the performance and energy-efficiency across various Hadoop applications.

I. Introduction

Low power is one of the main constraint for design of battery-operated embedded systems. However, this design objective has come into attention for high performance and data center systems as well. The main reasons are power constraint of the processor and physical constraint of the chip as the semiconductor industry has reached its physical scaling limits. In fact continues increase in the number of transistors on a chip has led to the so called "dark silicon" phenomena, where the power density does not allow all the transistors to turn on simultaneously [1]. As a consequence, hardware design paradigm has shifted from a performance-centric to an energy-efficient-centric methodology to respond to this challenge.

The energy demand of data centers that support big data computing ecosystems such as Hadoop MapReduce is increasing rapidly [2,3] which is the main obstacle for their scalability. Moreover, since energy consumption in data centers contributes to major financial burden, designing energy-efficient data centers is becoming very important. Current server designs, based on commodity high performance processors are not the most energy-efficient way to deliver green IT in terms of performance/watt [4,6]. Therefore, the embedded processors that are designed and developed based on energy efficiency metrics are finding their way in server architectures. Several companies have developed microservers based on ARM or Intel Atom cores. Due to the wide adoption of x86-based architectures in servers, we choose Atom to study, as it has a low power embedded micro-architecture with high performance x86 ISA.

In this work, we investigate system and architecture level parameters to optimize energy efficiency of the Big Data applications running on microserver employing low power embedded cores. Numerous big data applications rely on using the Hadoop MapReduce framework to perform their analysis on large-scale datasets. The tuning of Hadoop configuration parameters is vital for performance and power optimization of the running applications.

This paper in brief makes the following contributions:

- We analyze the impact of various tuning parameters at the system (number of mappers running simultaneously per microserver node and data block size), and architecture (operating voltage and frequency) levels on the performance, power and energy efficiency of Hadoop micro-benchmarks.
- We analyze how the interplay of various tuning parameters at system and architecture levels affects Hadoop applications power and performance sensitivity.

II. Experimental Setup

We conduct our study on Intel Atom C2758 server that has 8 processing cores per node and two levels of cache hierarchy. We test the performance of Hadoop representative micro-benchmarks including WordCount, Sort, Grep and TeraSort on Hadoop 1.2.1. Although Hadoop exploits cluster-level infrastructure with many nodes for processing big data applications, the focus of this paper is on single node performance and power to understand how various optimizations at the node level affects the energy-efficiency. We use *Perf* to capture the performance characteristics of the studied benchmarks. For measuring power dissipation of the microserver, Wattsup PRO power meter is used [5]. The studied configuration parameters are the number of mappers (1, 2, 4 and 8) equal to the number of active cores that runs simultaneously by task tracker, data block size (32MB, 64MB, 128MB, 256MB, 512MB) and operating clock frequency (1.2GHz, 1.6GHz, 2.0GHz and 2.4GHz).

III. Experimental Results And Analysis

In this section, we discuss the performance and energy-efficiency characteristics of Hadoop micro-benchmarks on microserver with respect to the number of mappers, data block size and operating frequency.

A. Execution time analysis

Figure 1 (represented as a bar graph) shows the execution time of the studied Hadoop applications. In all applications, the data block size of 32MB shows to have the highest execution time. The performance improves significantly with the increase in the data block size. This behavior is consistent across all the applications when the number of mappers is less than 4. In contrast, the wordcount shows a parabolic behavior at a large number of mappers and achieve the minimum execution time at the 256MB. *Sort* and *TeraSort* optimal data block size is 512MB whereas *WordCount* and *Grep* optimal block size is 64MB and 256MB with the maximum number of mappers. The general observation is that, the optimal data block size to maximize performance is closely decided by the application type and other tuning parameters such as number of mappers.

In addition, we have studied the impact of CPU frequencies to understand how Hadoop applications are sensitive to freque-

978-1-5090-1953-3/16/$31.00 ©2016 IEEE

ncy scaling. Sort application is less sensitive to the frequency, compared to other benchmarks. For this benchmark when CPU frequency is reduced to half the performance only drops by 20%. *Sort* is an I/O bound benchmark, which spends most of the time requesting data and waiting for I/O operations to be completed. In addition, as shown, increasing the number of mappers from 1 to 8 reduces the execution time.

B. Energy-Efficiency analysis

In order to characterize the energy efficiency, we evaluate Energy Product Delay (EDP) metric to investigate trade-off between power and performance when tuning Hadoop and processor parameters, as shown in Figure 1 (represented as line graph). We have observed that the increase in the number of mappers running simultaneously equal to the number of available cores, minimizes the EDP. Worst EDP is reported with one mapper, while 8 mappers gives the best EDP by effectively utilizing all available cores. The margin of EDP improvement becomes smaller with the increase in the number of mappers. The general observation is that the optimal energy efficiency is achieved when we utilize all available cores. In other words, the performance improvement achieved by adding more cores outweighs the power overhead associate with additional cores. Investigating the effect of the data block size on EDP, we observe that 128MB and 256MB is the optimal data block size for the *Word-Count* and *Grep* applications, however *Sort* and *Terasort*– I/O bound applications- achieves the optimal EDP with the largest data block size, namely 512MB, at the maximum number of mappers. The EDP trend is consistent with the execution time trend showing that in I/O bound applications, the maximum energy efficiency is achieved with the largest data block size. Moreover, we have conducted the analyses of frequency scaling on the EDP results. Energy efficiency is maximized at the highest frequency of 2.4GHz in all applications with the exception of *Sort*. *Sort* operating at a frequency of 1.6GHz provides the maximum energy efficiency as opposed to 2.4GHz frequency. As it was discussed earlier, *Sort* is an I/O bound application that spends a significant amount of execution time reading data from and writing it to HDFS. This behavior makes the performance of *Sort* almost insensitive to the operating frequency.

Another interesting observation is regarding the tuning of the data block size and frequency for various number of mappers. The results show that by simultaneously fine-tuning data block size and operating frequency we can reduce the number of mappers and yet be as energy-efficient as with the maximum number of mappers. For example, Grep of 512MB block size and 2.4 GHz frequency with 2 and 4 mappers achieves higher or similar energy-efficiency compared to maximum number of mappers. This indicates that in the absence of available cores, we can fine-tune frequency and block size with fewer mapper and still be as energy-efficient competitive with more number of cores/mappers.

IV. KEY FINDINGS

- Default Hadoop configuration parameters are not optimal for maximizing the performance and energy-efficiency.
- The speedup obtained when increasing the number of available cores on microserver node outweighs the power overhead associated with increasing the number of cores.
- Although utilizing all available cores on each microserver node provides maximum energy-efficiency across the studied applications, concurrent fine-tuning of frequency and data block size reduces the reliance on the maximum number of cores. This would help freeing up cores on each node to accommodate co-scheduling incoming applications without sacrificing the energy-efficiency.
- I/O bound applications can be run at a lower frequency to save power. Performance loss can be compensated to a significant extend by increasing the number of mappers.

REFERENCE

[1] B. Raghunathan, et al. "Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors." Proc. of DAC, 2013.

[2] M. Malik, et. al., "Big data on low power cores: Are low power embedded processors a good fit for the big data workloads?." Proc. of ICCD 2015

[3] L.A. Barroso, et al., "The datacenter as a computer: An introduction to the design of warehouse-scale machines."Synthesis lectures on computer architecture 2013.

[4] K. Neshatpour, et al. "Energy-efficient acceleration of big data analytics applications using FPGAs." Proc. of Big Data 2015

[5] Blem, Emily, et.al. "Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures " Proc. in HPCA2013.

[6] M. Malik, et al. "System and architecture level characterization of big data applications on big and little core server architectures." Proc. of Big Data 2015

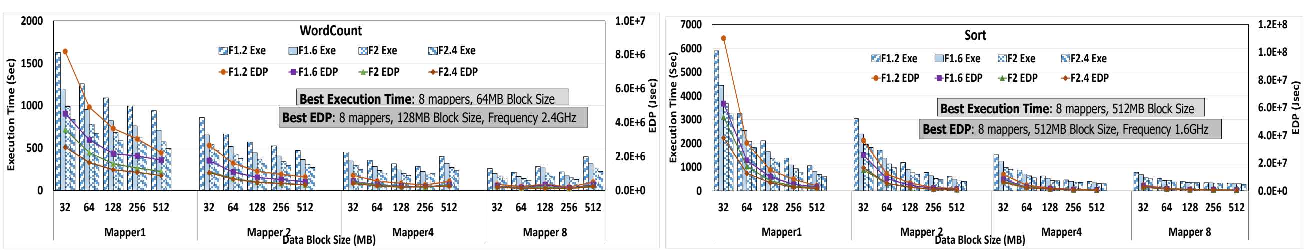

Figure 1. Execution Time and EDP of (a) Wordcount (b) Sort with various mappers, data block size and operating frequencies

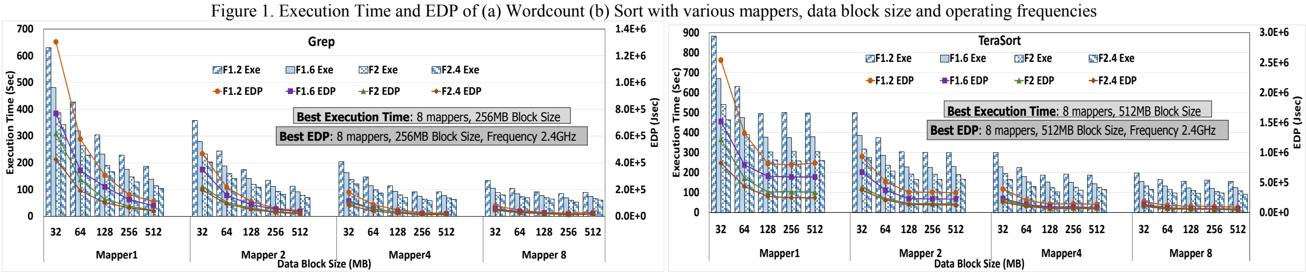

Figure 1. Execution Time and EDP of (c) Grep (d) Terasort with various mappers, data block size and operating frequencies